INTEGRATING BIO-MECHANICS AND DEEP LEARNING FOR YOGA POSE CLASSIFICATION LEVERAGING JOINT ANGLE ESTIMATION

A SYNOPSIS

Submitted by

J.PALANIMEERA
(Reg. No. 9200017002)

in partial fulfillment for award of the degree

of

DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER

APPLICATIONSSCHOOL OF COMPUTING

KALASALINGAM ACADEMY OF RESEARCH AND EDUCATION

(DEEMED TO BE UNIVERSITY)

ANAND NAGAR, KRISHNANKOIL – 626126

TAMILNADU, INDIA

SEPTEMBER 2023

1. INTRODUCTION

In the field of computer-vision, Human Activity Recognition (HAR) is a major discussion. This is attributed to the exponential growth of video recordings and the large number of applications that use scheduled video testing, such as visual surveillance, human-machine interfaces, video interrogation, and video recovery. Human activity recognition, especially abnormal state action recognition, is one of the most appealing of these applications. A function is a continuum of development of the human body involving several body parts simultaneously. Action recognition is the computer's incorporation of an idea (such as a video) with pre-categorized types. After that, send it to an action-style label. Human movements are often divided into four phases in terms of diversity: gestures, behavior, interactions, and group activities [1] as shown in Figure 1, the essential movement of physical body parts is described as a gesture. Some examples include handshaking, and facial expressions. Within the defined forms, a gesture usually takes a really short time and features a very low level of complexity. An action is a type of action that an individual engages in. In reality, it's a mash-up of various gestures (nuclear actions). Running, biking, jogging, and boxing are some examples of acts. Communication is a type of interaction in which two characters interact. It may be a human or object encounter. Human interaction includes fights, handshakes, and embraces between two people, while object communication includes an ATM user, a computer user, and a bag thief. The most complicated form of function is the team function. It can, of course, be a mix of movements, acts, and interactions. It includes more than two people as well as one or more things. Team action may take the form of a protest, a game between two teams, or a team meeting and much study has been done following the basic development of human motion recognition. Feature extraction, process learning, classification, process identification, and division are all important aspects of such a structure [2].

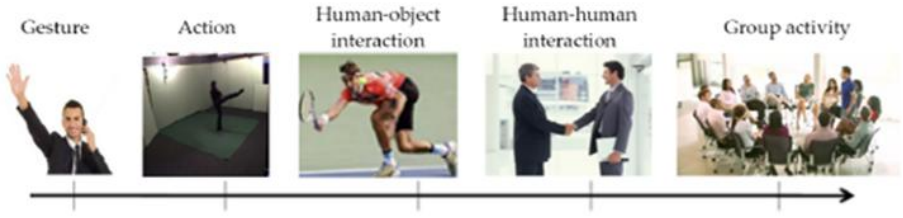


Figure.1.Categorization for different level of activities

The problem with yoga however is that, just like any other exercise, it is of utmost importance to practice it correctly as any incorrect posture during a yoga session can be

unproductive and possibly detrimental. This leads to the necessity of having an instructor to supervise the session and correct the individual's posture. Since not all users have access or resources to an instructor, an artificial intelligence-based application might be used to identify yoga poses and provide personalized feedback to help individuals improve their form [3].In recent years, human pose estimation has benefited greatly from deep learning and huge gains in performance have been achieved [4]. Deep learning approaches provide a more straightforward way of mapping the structure instead of having to deal with the dependencies between structures manually. The [5] proposed a deep learning to identify 5 exercise poses: pull up, swiss ball hamstring curl, push up, cycling and walking. However, using this method for yoga poses is a relatively newer application [3].

The [6] proposed a 'Yoga Tutor' project that uses accelerated robust characteristics to make the distinction in postures between a learner and a professional (SURF).The author [7] presented a picture and text-based intelligent systems for yoga, but they did not look at the posture of the practitioner. The author [8] used a features-based method to create a self-training system that recognized yoga exercises. It makes use of a Kinect to capture the person's body contour and create a body map. To obtain a descriptor for the human position, a star skeleton was employed for quick skeletonization. In reference [9], a yoga identification system that is based on a Kinect and the AdaBoost classification with a 94.78 accuracy score is proposed for six asanas. They are, however, utilizing a depth sensor-based camera that is normally not obtainable. Using convolutional neural network (CNN) and stacked autoencoder (SAE) methods, Mohanty et al. [10] implemented an image recognition approach for detecting Indian traditional dance and yoga postures from photographs. They did, though, only analyze their competence on still photos, never on videos. Since the introduction of DeepPose by [11], the traditional skeletonization methodology has been supplanted by deep learning-based technologies. DeepPose is leading the charge away from traditional techniques and toward deep network-based approaches. It directly regresses on joint coordinates using deep neural network-based regressors. It anticipates a person's activities and also forecasts the location of hidden body parts. However, their approach has difficulty with localization.

This research focuses on examining several methods for classifying yoga poses and aims to gain understanding of posture estimation and the real-time application of deep learning to yoga pose classification.

1. HISTORY OF YOGA

Humans are prone to musculoskeletal disorders with aging and accidents [12]. In order to prevent this some, form of physical exercise is needed. Yoga, which is a physical and spiritual exercise, has gained tremendous significance in the community of medical researchers. Yoga has the ability to completely cure diseases without any medicines and improve physical and mental health [13]. A vast body of literature on the medical applications of yoga has been generated which includes positive body image intervention, cardiac rehabilitation, mental illness etc. [13]. Yoga comprises of various asanas which represent physical static postures. The application of pose estimation for yoga is challenging as it involves complex configuration of postures. Furthermore, some state-of-the-art methods fail to perform well when the asana involves horizontal body posture or when both the legs overlap each other. Hence, the need to develop a robust model which can help popularize self-instructed yoga systems arises.

2. LITERATURE SURVEY

As people age and have accidents, they are more susceptible to musculoskeletal disorders. It is necessary to engage in physical activity to prevent this. Yoga is a physical and spiritual practice that has grown significantly in significance among scientists who study medicine. Without the use of medication, yoga has the power to completely cure illnesses and enhance both physical and mental health. A substantial body of literature has grown on the clinical uses of yoga, including psychiatry, cardiac rehabilitation, and positive body image intervention. In yoga, there are many asanas that stand for firm body positions. Yoga has a complex configuration of postures, making it difficult to apply pose assessment to it. Additionally, when the asana requires a horizontal body posture or when both legs are crossed, some sophisticated techniques do not function properly. Therefore, it becomes necessary to create a solid model to aid in the spread of self-instructional yoga systems.

2.1 Survey on Human Pose Estimation

Human posture recognition has made huge advancements in the past years. It has evolved from 2D to 3D pose estimation and from single person to multi person pose estimation. [13] uses pose estimation to build a machine learning application that helps detect shoplifters whereas uses a single RGB camera to capture 3D poses of multiple people in real-time. Human pose estimation algorithms can be widely organized in two ways. Algorithms prototyping estimation of human poses as a geometric calculation are classified as generative methods while algorithms modelling human pose estimation as an image processing problem are classified as discriminative methods [14]. Another way of classifying these algorithms is

based on their method of working. Algorithms starting from a higher-level generalization and moving down are called top-down methods, whereas algorithms that start with pixels and move upwards are called bottom-up methods [15].

3.1.1. Generative

Generative procedures provide a technique to predict the features from a given pose hypothesis. They start with initializing the posture of the human body and project it to the image plane. Adjustments are made to make the projected image and current image observations compliant. Generative based approaches offer easy generalization due to less constraint of a training pose dataset [16]. However, due to the high dimensional projection space search, this method is not considered computationally feasible, and is thus slower as compared to discriminative methods. [17] describes a generative Bayesian method to track 3D segmented human body figures in videos. This is a probabilistic method which consists of a generative model for image appearance, an initial probability distribution over joint angles and pose that represents movement of humans and a robust likelihood function. Even though the method is able to track humans in unknown complicated backgrounds, it faces the risk of eventually losing track of the object.

3.1.2. Discriminative

Contrary to generative methods, discriminative methods start with the evidence of the image and learn a technique to model the relationship between the human poses and evidence on the basis of training data. Model testing in discriminative methods is a lot faster as opposed to generative methods due to the search in a constrained space as opposed to a high dimensional feature space [18]. [19] explores a discriminative based learning method to obtain 3D human pose from silhouettes. This approach does not require a body model explicitly nor any prior labelled parts of the body in the image. It restores the pose using non-linear regression based on the shape descriptor vectors fetched automatically from silhouettes of images. It uses Relevance Vector Machine (RVM) regressors and damped least squares for regression. The method, though increasing the accuracy by three times, is not accurate enough, as there are some instances of incorrect poses and results showing significant temporal jitter. Discriminative methods are further categorized into learning methods and exemplar methods [19].

A. Learning based – deep learning:

One important learning-based method is deep learning which is built upon Artificial Neural Networks (ANNs). ANN is analogous to the human brain where the units in an ANN represent the neurons in the human brain, and weights represent the strength of connection

between neurons. Deep learning provides an end-to-end architecture that allows automatic learning of key information from images. One popular deep learning model which has been widely used for pose estimation is Convolutional Neural Network (CNN) which will be discussed later. [20] have contributed to the research by using CNNs and stacked auto-encoder algorithms (SAE) for identifying yoga poses and Indian classical dance forms. However, their performance evaluation is done only on images and not on videos.

B. Exemplar methods:

In exemplar methods, pose estimation is based on a unique set of poses with their equivalent representations [19]. Classification algorithms such as random forests and randomized trees are robust and fast enough to manage this. Random forest is made up of various randomized decision trees and is hence called an ensemble classifier. It consists of non-terminal nodes which have a decision function to predict the similarities in images. [21] used an improved version of random forests which comprised of two levels of random forests. The first layer of the tree acted as a discriminative classifier to classify body parts and passed the classification results to the second layer which predicted joint locations in the body. Another approach similar to random forests is Hough forests [19] which consists of decision forest combinations, where the terminal node in every tree is either a regression or classification node. Enhanced Hough trees have a parts objective ("PARTS") which is an optimized objective based on discrete information gain.

C. Top- down:

Most studies refer to generative methods as top-down methods [19]. The top-down approach obtains keypoints by using a module to detect human subjects to which a pose estimator can be applied. The primary advantage of top-down methods is their ability to break down the task into multiple smaller tasks. These smaller tasks include detecting the object followed by pose estimation. For this, the detector should be powerful enough to detect hard or comparatively smaller objects so that the performance of the pose estimator is improved. [22] uses a top down approach to articulate human pose estimation and tracking. Their top down approach comprises of three components – a human candidate detector, a single person pose estimator and a human pose tracker. A general object detector is chosen to identify human candidates after which a cascaded pyramid tracker is used to recognize the equivalent human pose and lastly, a flow-based pose tracker is used to assign a distinct and temporally consistent id to each human candidate for multi pose tracking. Even though this research develops a modular system to human pose estimation and tracking, they achieve an average precision of only 69.4 for pose estimation.

D. Bottom – up:

This approach involves detection of human keypoints from potential subjects and organizing them into human limbs using several data association mechanisms. The cost of computation in this method does not depend on the number of human subjects in the images. Thus, it provides a very good compensation between cost and accuracy. Some surveys refer to bottomup methods as discriminative methods [19]. [23] proposes a novel approach that combines traditional bottom-up and top-down methods for multi-person pose estimation. The feed forwarding to the network is done in a bottom-up fashion while parsing of poses along with bounding box constraints is performed in a top-down fashion. The features from the image are extracted using a residual network which is trained to learn the confidence map of the joints and relationships of connectivity between the joints. This residual network is nothing but ResNet50, which is a deep neural network architecture. Although this research does not focus on classifying the poses, it shows how deep learning architectures like ResNet50 can be used in order to make human pose estimation accurate.

3.2. Survey on Keypoint Detection Methods

3.2.1. Openpose

OpenPose is a multi-person real-time keypoint detection which brought a revolution in the field of pose estimation. It was invented in Carnegie Mellon University (CMU) by the Perceptual Computing Lab[13]. It uses CNN based architecture to identify facial, hand and foot keypoints of a human body from single images. OpenPose helps identify human body joints using an RGB camera. OpenPose keypoints include eyes, ears, neck, nose, elbows, shoulders, knees, wrists, ankles and hips. It presents the results obtained by processing inputs from a camera in real-time or pre-recorded videos or static images as 18 simple keypoints. Because of this, it finds its use in a variety of applications ranging from sports, surveillance, activity detection to yoga pose recognition. The work proposed in [18] uses OpenPose for initial keypoint identification followed by CNN for classification of yoga poses. However, they achieve an accuracy of only 78% which could be due to the limited dataset they used or architecture and hyperparameter tuning of their CNN model.

3.2.1. Posenet

PoseNet is another deep learning framework similar to OpenPose which is used for identification of human poses in images or video sequences by identifying joint locations in a human body. These joint locations or keypoints are indexed by "Part ID" which is a confidence score whose value lies in the range of 0.0 and 1.0 with 1.0 being the greatest. The PoseNet model's performance varies depending on the device and output stride [14]. The

PoseNet model is invariant to the size of the image, thus it can predict pose positions in the scale of the actual image irrespective of whether the image has been downscaled.

3.2.3. Pifpaf

PifPaf is a new method based on the bottom-up approach for 2D multi-person human pose estimation. It uses a Part Intensity Field (PIF) for body part localization and a Part Association Field (PAF) for association of body parts to form full human poses [15]. The model beats other methods in terms of a lower resolution and better performance in overcrowded places mainly due to the following: (a) fine information encoded in a newer composite field PAF, (b) the selection of Laplace loss that integrates an opinion of uncertainty. The model architecture rests upon a completely convolutional box-free design [22] .

3. PROBLEM STATEMENT & OBJECTIVES

3.1 Problem Statement

In today's era, people are increasingly concerned about physical health and fitness. Exercise is one of the most effective ways to maintain good physical health. However, many people lack the knowledge and guidance required for performing exercises correctly, leading to incorrect posture and ultimately increasing the risk of injury. This research aims to help yoga teachers monitor the movement of yoga practitioners, which can detect the user's pose during exercise and provide feedback on the correctness of the posture, which can help prevent injuries and improve the user's form.

The problem addressed in this research is to develop an integrated framework for pose estimation using artificial intelligent techniques to aid people in performing exercises with correct posture. The primary objective is to prevent injuries and improve the form of human exercise through a computer and camera. The proposed system uses a modified open pose, which is a novel approach for posture detection and correction. This system integrates deep learning models such as LSTM. The research also aims to provide personalized feedback on yoga poses using pose evaluation and the output of deep learning algorithms. The performance of the system is evaluated with respect to the speed and accuracy of the pose estimation framework using open poses on a real-time dataset.

4.2. Objectives

The main aim of this research is to aid people in performing the correct posture for exercises by building Personal Exercise Assistance that detects the user's exercise pose and provides valuable feedback on the user's form, using a combination of the latest advances in Pose estimation and Deep learning. The following are the mentioned objectives.

- **Objective 1:** To Predict Yoga Pose Using Deep Learning Algorithm
- **Objective 2**: To Classify Yoga Posture In Time Series Using A Deep Learning Algorithm
- **Objective 3:** To Recognize Yoga Posture Using A Transfer Learning Approach
- **Objective 4:** To Find Joint Angle Values In Real Time Using Deep Learning Algorithm.

4. PROPOSED METHODOLOGY

Deep learning is widely used for image classification tasks wherein the model takes input in the form of images and outputs a prediction. Deep learning algorithms use neural networks to determine the connection between the input and output.

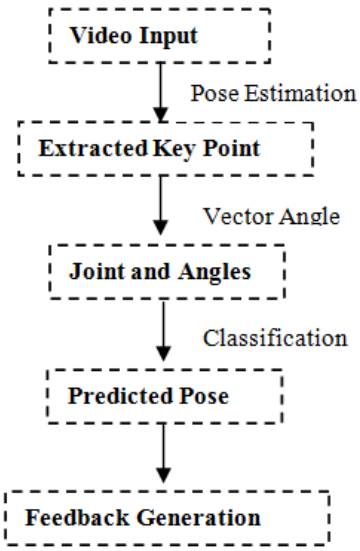


Figure.2. The overview of the proposed method.

For pose estimation problems, the image with pose of individuals is taken as input and the deep learning model tries to learn correctly the different poses so as to accurately classify them. As one can guess, this could be a computationally expensive task if the number of images is large. Also, as we want accurate results we would not want to compromise on the quality of the images as that could affect the features extracted by the model. Thus, in this work we propose using OpenPose (a pretrained model) to extract keypoints of the human joint locations from the images and then training the deep learning model on these keypoints.

Finally, to detect correct yoga poses and provide feedback to improve the yoga posture. The Figure 2 illustrates the overview of the proposed method.

5.1. yoga asana prediction using deep learning algorithm

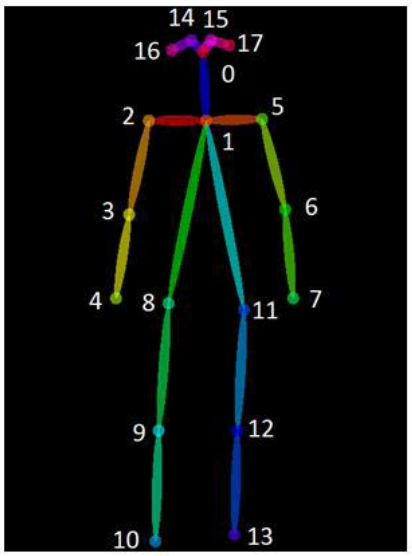
Our methodology anticipates that the client will intuitively feel yoga postures based on real time and pre-recorded videos. The method can be divided into four fundamental progressions. First, information classification is carried out, using either the most recent cycle connected to an identity or recently pre-recorded videos. Following bilateral integration and segmentation, pose estimation is used to separate joint regions using region confidence maps and region correlation fields. The key points are fed into our model, which uses pose estimation to find examples and LSTM to detect changes over time. Long-term research is being done on the model and method, as well as framewise anticipation and survey approach preparation for 45 edges (1.5 seconds) of result. The proposed technique detects participants' yoga practice in real-time and automatically records videos.

The process can be divided into four basic steps.

- The initial stage is data collection, which can be done using a real-time approach to detection or in conjunction with previously recorded videos.
- The next step determines joint positions using the pose estimation technique.
- Then, the LSTM model analyses the identified key points' changes over time.
- Finally, yoga action is recognized from real-time video.

5.1.1. Yoga Asana Posture Extraction

This is the first step in the process, and it makes use of pose estimation. This stage occurs offline in the case of recorded movies, while it occurs online in the case of real-time predictions, with input from the camera providing key points about the suggested model. Open Pose is an opensource toolkit for multi-person key point identification that simultaneously identifies key points on the human structure, hands and face. Figure.3. Shows the locations of 18 key points monitored by the Open-Pose, including ears, eyes, nose, neck, shoulders, hips, knees, ankles, elbows and wrists. Every single video frame is produced in JSON format file, which comprises the positions of each body component for every individual recognized in the picture.



Nr	Name
0	Nose
1	Neck
2	RShoulder
3	RElbow
4	RWrist
5	LShoulder
6	LElbow
7	LWrist
8	RHip
9	RKnee
10	RAnkle
11	LHip
12	LKnee
13	LAnkle
14	REye
15	LEye
16	REar
17	LEar

Figure.3. The position of human body joints using the pose estimation

The pose extraction was done at the Open Pose network's default resolution for best results. At these settings, the system averaged approximately 3 frames per second. The process flow for identifying real-time human actions is shown in Figure 4.

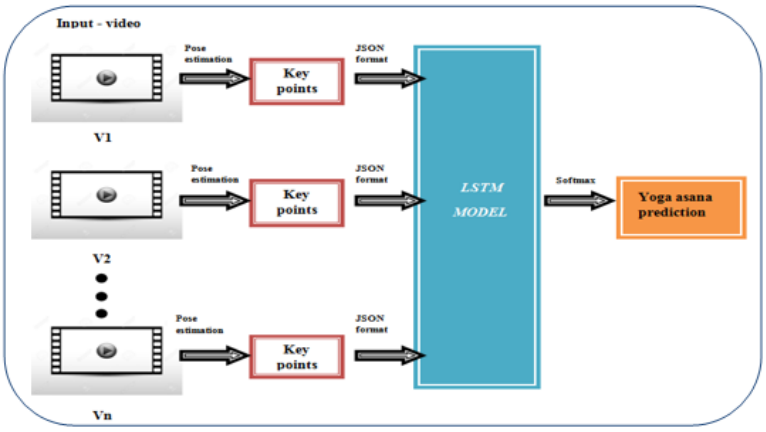


Figure.4. Flowchart for the workflow

5.1.2. Prediction for Real-Time Yoga Asana

Predictions are generated in real-time for ten people (5 men and 5 females) using a 1080p HD RGB camera on a machine with an Intel® CORE™ i5-5200U CPU and 32 GB RAM. Data gathering and real-time prediction were made by a variety of people. In an indoor setting, all participants performed asana at a gap of 4 meters from the camera. To identify whether the user was executing one of the yoga asanas or not, thresh holding on Softmax output was used. Open Pose ran at about 5 frames per second on the machine, and it took around 5 seconds to create our model with 45 input frames. As a result, the forecast made at some given time is for the preceding 5 seconds. The user can alter the net resolution setting of

Open Pose to get faster but less accurate results. Table 1 summarizes the outcomes in real-time. The equation below describes the accuracy calculation method.

Accuracy = (CC (Correct cases) / TC (Total case)) * 100.

The method achieves a high level of accuracy of 98 percent, with individual asana accuracy above 96 percent. These findings are consistent with those from the test set, indicating a positive outcome in real-world settings.

Table 1 summarizes the outcomes in Real-Time

S.no	Yoga asana	Total case(TC)	Correct case(cc)	Accuracy %
1	Bridge	120	116	96.2
2	Child's	124	120	98
3	Downward Dog	132	130	97.35
4	Mountain	140	136	96
5	Plank	135	132	97.3
6	sitting forward bend	133	130	98.6
7	Tree	125	125	99.2
8	Triangle position	135	130	98.2
9	Warrior1	120	120	96.26
10	Warrior2	120	120	98.26

5.1.3 Experimental analysis

The machine achieves 99 percent accuracy on training data and 99 percent accuracy on validation data after 100 epochs of training. For each frame, the machine achieves a test accuracy of 99 percent. Figure 5 depicts model accuracy, model loss. The confusion matrix for predictions using the proposed approach is shown in Figure 6.

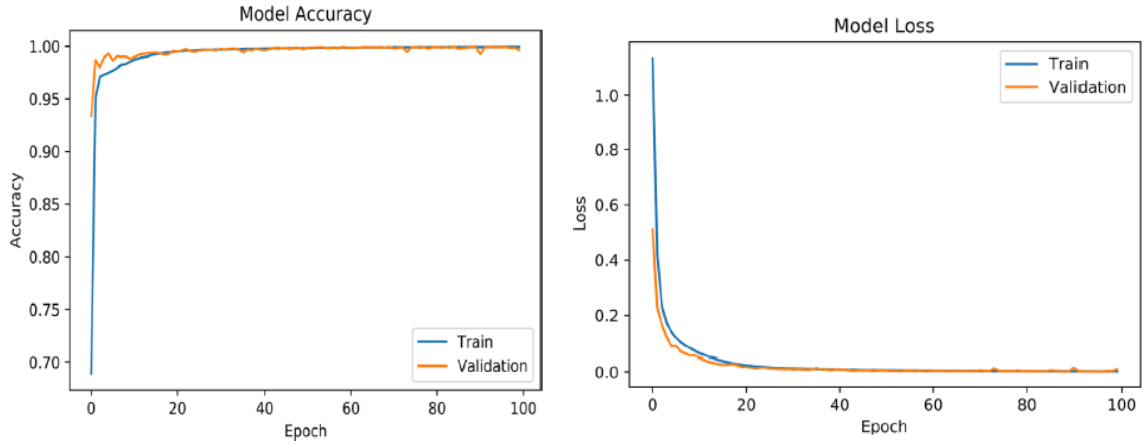


Figure.5. Depicts model accuracy, model loss.

The high density on the diagonal indicates that most activities were properly predicted.Except for Mountain posture for test cases, the execution is considerably above and close to flawless

in most of the Yoga postures.Mountain posture was categorized as Plank 1000 times out of 20000 frames,resulting in 94 percent accuracy. Similarly, a plank is sometimes mislabeled as a mountain. The explanation for this discrepancy might be that both asanas are executed while standing and have comparable starting phases of development. Figure 7 depicts real-time yoga asana prediction and yoga asana prediction in recorded videos.

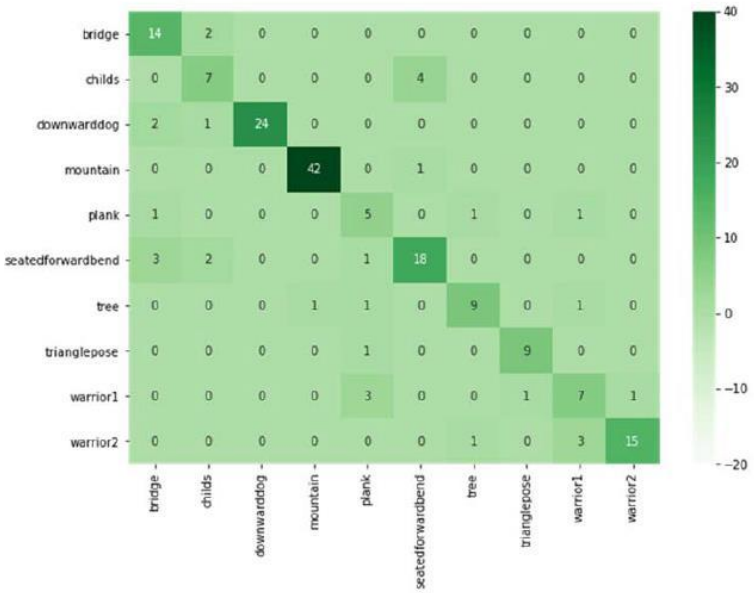

bridge 14 2 0 0 0 0 0 0 0 0
childs 0 7 0 0 0 4 0 0 0 0
downwarddog 2 1 24 0 0 0 0 0 0 0
mountain 0 0 0 42 0 1 0 0 0 0
plank 1 0 0 0 5 0 1 0 1 0
seatedforwardbend 3 2 0 0 1 18 0 0 0 0
tree 0 0 0 1 1 0 9 0 1 0
trianglepose 0 0 0 0 1 0 0 9 0 0
warrior1 0 0 0 0 3 0 0 1 7 1
warrior2 0 0 0 0 0 0 1 0 3 15
bridge
childs
downwarddog
mountain
plank
seatedforwardbend
tree
trianglepose
warrior1
warrior2
40
30
20
10
0
−10
−20

Figure.6.Confusion matrix for identifying using the proposed approach

Figure. 7. Real-time yoga asana predictions

5.2 **Classify Yoga Posture In Time Series Using A Deep Learning Algorithm**

The goal of this research is to create a safer and more comfortable setting for solo yoga practitioners. Figure 8 depicts the framework of the suggested technique.

• For the recognition of skeletal-based yoga postures, a DL model based on Long-Range modelling is proposed; this model captures the global aspect and local motion dynamics of the yoga posture while combining the visual appearance and Long-Range temporal dynamics.

• Using a sparse sampling method, a succinct abstraction is made beneficial toget long-term temporal correlations. The SRL now has better global and dynamic local attributes.

• For obtaining long-range temporal patterns in both forward and reverse directions, a bi-directional LSTM network (Bi-LSTM) is given. It substantially facilitates gradient vanishing, enhances feature propagation, encourages feature reuse, and drastically reduces the number of parameters.

• The suggested Bi-LSTM model beats cutting-edge algorithms for posture identification, according to simulations that use open pose data to extract crucial points from real-time filmed yoga posture films.

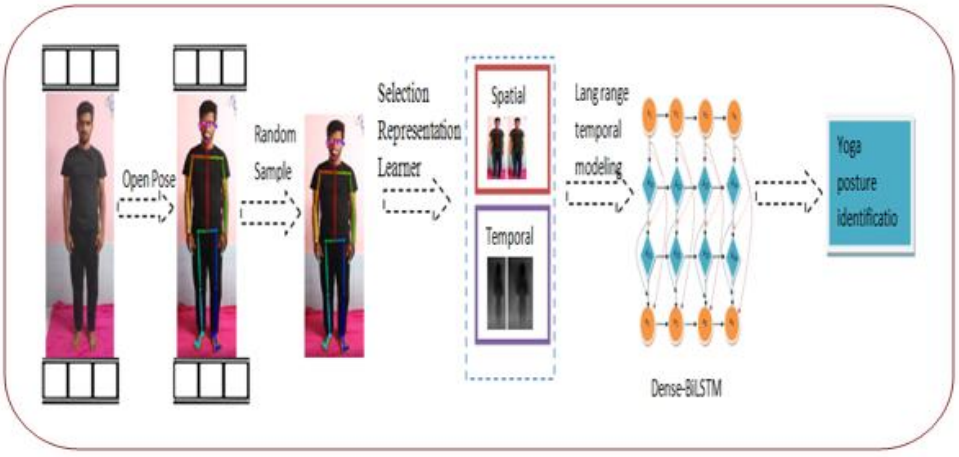


Figure.8 depicts the framework of the suggested technique

5.2.1 Stack-Representation Learner

Selection level and representative learning, which make up SRL, are intended to extract long-range transitory patterns. Representation the selection layer is used to build a layered structure as the learner strives to capture the long-range temporal connection that gives the layer's characteristics their shape. Yoga poses are often integrated with a series of video scenarios, most of which reflect the core theme of activities. However, the beginning and finish of a video are more likely to have scene altering and transition segments that are unrelated to the skeleton action. Because videos include redundant information, sampling

numerous frames or clips from videos is a feasible method for action recognition. A sample stack is made by randomly gathering frames from fixed-length video segments and chaining them together to create a new collection of segments that balances efficiency and effectiveness. Only one image from the sampler stack is chosen at random for each section. The sequence of segments is then used to construct a stack of frames.

In particular, given an n-frame video,

$$V = \{i_o , \dots i_i, \dots i_{i-1}\}$$

Where, where i_i denotes the video's i_{th} frame Video v is separated into k segments, each of which is M= n/k in length.

One selection frame Sf_i is chosen at random for each segment s_i . In the simultaneously, an optical -flow map Of_i is selected at random from five successive frames.

The Selection Representation Learner (SRL) is defined as:

$$Sample\ representation\ learner(s, w) = \boldsymbol{\omega}(\boldsymbol{g}(x_0,\text{w}),(\boldsymbol{g}(x_1,\text{w}),\boldsymbol{g}(x_2,\text{w}),\dots\ \boldsymbol{g}(x_{m-1},\text{w})$$

Where, $\boldsymbol{g}(x,\text{w})$ denotes the back-bone networks, x is the selection unit, which might be from the visual or temporal branches, and w denotes its weights.

5.2.2 Experimental analysis

Real-time yoga postures skeleton data performance is rather excellent. Obviously, the bulk of the categories have greater than 85% accuracy due to the efficacy of our algorithm. In Figure 9 depicts the confusion matrix. The spatial features are quite similar, and their variances in motion are little, which CNN may disregard.

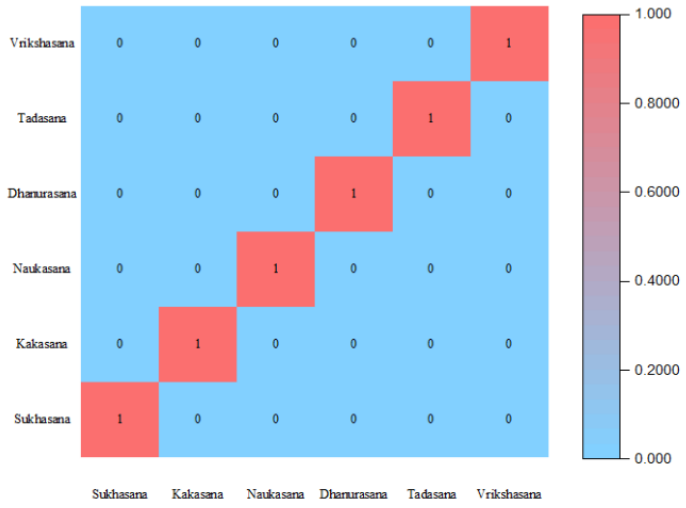
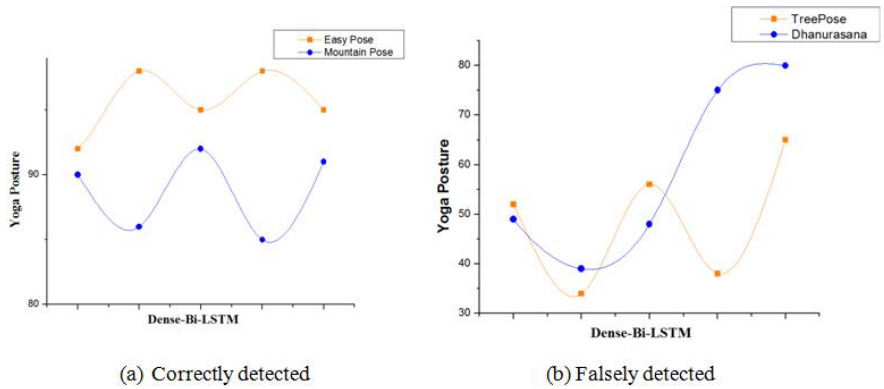


Figure.9.Confusion matrix

It can be understood that identifying fine-grain level yoga postures is still difficult for existing models. Figure 10 depicts four successfully recognized and two correct and two incorrectly detected Dense-BiLSTM cases and their accompanying detection scores.



(a) Correctly detected (b) Falsely detected

Figure.10. Proposed method and their corresponding detection yoga posture

5.3. Pose Recognition using Transfer Learning with Deep Representations

In machine learning, utilizing the previously learnt knowledge for solving a new task is known as transfer learning or knowledge transfer . The transfer learning using deep CNNs is very helpful for training the model with limited size dataset, because CNNs are prone to overfitting with small dataset. However, the overfitting can be avoided by increasing the size of the training data, but it is very difficult and expensive to provide the large amount of annotated data. In this situation, the transfer learning comes handy and solves this problem by using the pre-trained deep representation as a source architecture for building the new architecture.

5.3.1. Transfer Learning

Transfer learning, used in machine learning, is the reuse of a pre-trained model on a new problem. In transfer learning, a machine exploits the knowledge gained from a previous task to improve generalization about another. In this work, we have employed the AlexNet as a source architecture for solving human action recognition problem. The AlexNet was trained on ImageNet dataset and takes as input 224 x 224 pixels RGB image and categories it into the respected class. This architecture consists of five convolutional layers from C1-C5 and three fully connected layers Fc6-Fc8 as shown in the top row of the Figure. 11.

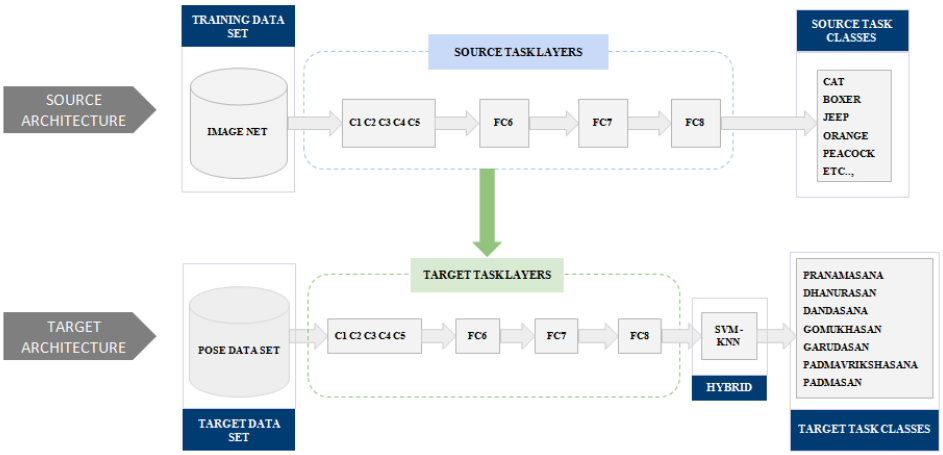


Figure. 11. Overview of the proposed system, first row indicates the source architecture and second row shows the target architecture.

5.3.2 Hybrid Algorithm (SVM-KNN)

SVM and KNN each have their own set of disadvantages; we may combine them to create a more powerful classification system. The separation between the testing and training data is determined using the first K closest neighbor algorithm. However, if they submit data that is noisy or contains irrelevant information, KNN becomes imprisoned and produces very erroneous conclusions. Almost every image classification approach calculates the distance between the test and training samples. KNN's main duty is to discover the query sample's neighbors first, and then categories it based on the majority class of its closest neighbors. The design of a distance measuring function is quite important, and it varies depending on the situation to which it will be used. The following is a general equation for calculating the distance between test and training samples.

$$QC' = \sum_{(t_i, c_i) \in K_d} s(c=c_i)$$

'c' stands for a class label, t_i for the training sample at ith position, 'c_i' for the 'i^{th}' nearest neighbors, and 'k_d' for the nearest neighbor list.

For classification, each feasible neighbor's training sample and class label is compared to the class labels. Then, to make a decision, majority voting might be used. However, if the procedure results in a blind or ambiguous conclusion, the current process will be switched to a Support Vector machine (SVM). SVM is machine learning algorithms that can categories pictures by constructing a hyper plane that divides one sample from another by a certain distance. When samples are significantly closer to the border, SVM performance suffers. To break one sample features class from another in SVM, the optimum hyper plane must be generated. The optimized hyper plane will span the highest distance margin and will be more

accurate than the other hyper planes in identifying feature data tuples.

The following phrase is used to obtain the optimum hyper plane.

$$wt+b = 0$$

The training samples must lie on hyper plane margin lines to be defined as support vectors. w – Vectors perpendicular to the separation hyper plane t – Set of training vectors b–Offset parameter that permits the margin to be increased.

5.3.3 Experimental analysis

Table 2. Comparison of classification results on Real time dataset

S No:	Methods	% Accuracy
01	Proposed method (SVM-KNN)	98.17
02	Proposed method (KNN)	94
03	Proposed method (SVM)	89
04	Charalampous and Gasteratos[12]	91.99
05	Ahad et al. [15]	86.7
06	Ding and Qu [16]	95.58
07	Chaaraoui et al. [17]	89.86

During experimentation, the dataset is divided into two parts, One part is used for training while other one is used for testing the correctness of the proposed method same as [18]. The proposed method achieves 98.17% accuracy on real time dataset, which is higher than the similar methods such as [12, 15, 16, 17], as shown in Table 2. The confusion matrix indicating the accuracy of each action and correspondence between the target classes along x-axis and output classes along y-axis is shown in Figure. 12. On a real-time yoga asana dataset, the proposed technique achieves 98 percent accuracy, shown in Figure 13 which is greater than previous methods such as other action-related

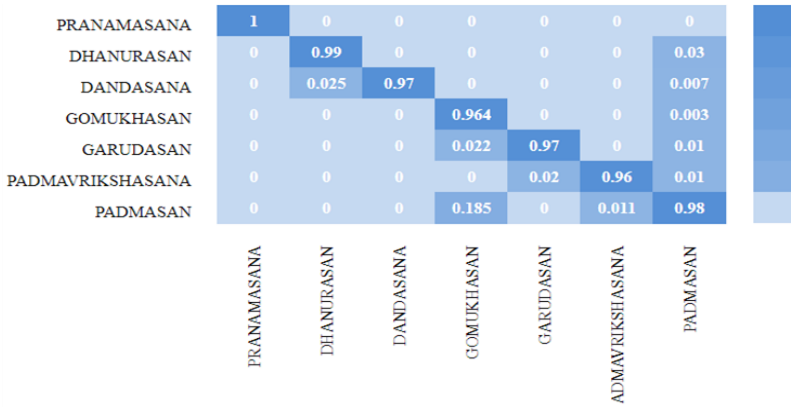


Figure.12. Show a confusion matrix that shows the correctness of each yoga asana

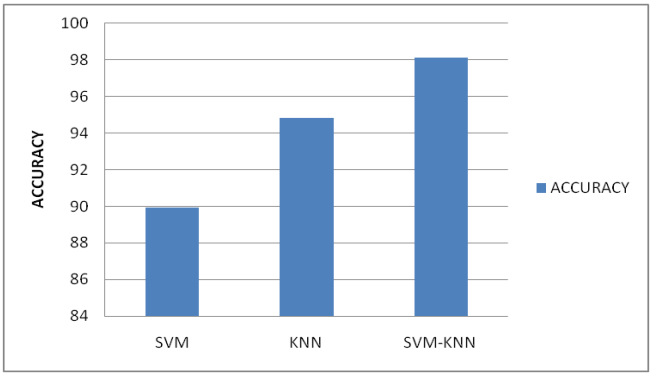


Figure.13.Comparison of classification results in action dataset

5.4. Bio vision Hierarchy Using Deep Learning Model

Bio vision, a motion capture service provider, created the Bio vision Hierarchical Data (BVH) file format in order to provide motion capture data to their customers. A BVH file is divided into two sections: a header section that describes the skeleton's hierarchy and initial pose, and a data section that contains the motion data. Examine the "Example1.bvh" example BVH file shown in Figure 14. The keyword "HIERARCHY" appears at the start of the header section. The line that follows begins with the keyword "ROOT" and is followed by the name of the root segment of the hierarchy to be defined. After this hierarchy is described, it is permissible to define another hierarchy; this too would be denoted by the keyword "ROOT". In concept, a BVH file can have an unlimited number of skeleton hierarchies. The format of the motion section limits the number of segments in practise; one sample in time for all segments is on one line of data.

```
HIERARCHY
ROOT Hips
        {
                OFFSET 0.00 0.00 0.00
                CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
                JOINT Chest
                {
                        OFFSET 0.000000 6.275751 0.000000
                        CHANNELS 3 Zrotation Xrotation Yrotation
                        JOINT Neck
                        {
                                OFFSET 0.000000 14.296947 0.000000
                                CHANNELS 3 Zrotation Xrotation Yrotation
                                JOINT Head
                                {
                                        OFFSET 0.000000 2.637461 0.000000
                                        CHANNELS 3 Zrotation Xrotation Yrotation
                                        End Site
                                        {
                                                OFFSET 0.000000 4.499004 0.000000
                                        }
                                }
                        }
```

Figure.14. BVH motion data

5.4.1. Long Short Term Memory

LSTM methods were used to create the tool's prototype in Python utilising Keras(https://keras.io) and Tensor Flow(https://www.tensorflow.org). The simplicity of these tools, which made them more appropriate for quick prototyping, was a major factor in this decision. The solution utilised in these studies was made up of three layers, the first of which is a 15-neuron LSTM layer (input layer). This was limited to the 10 yoga poses used in this study to tag Motion-Capture events to form a 3D animation. These poses were essential since the neural network must be trained to recognise each class's unique set of BVH files, which must be accurately constructed.

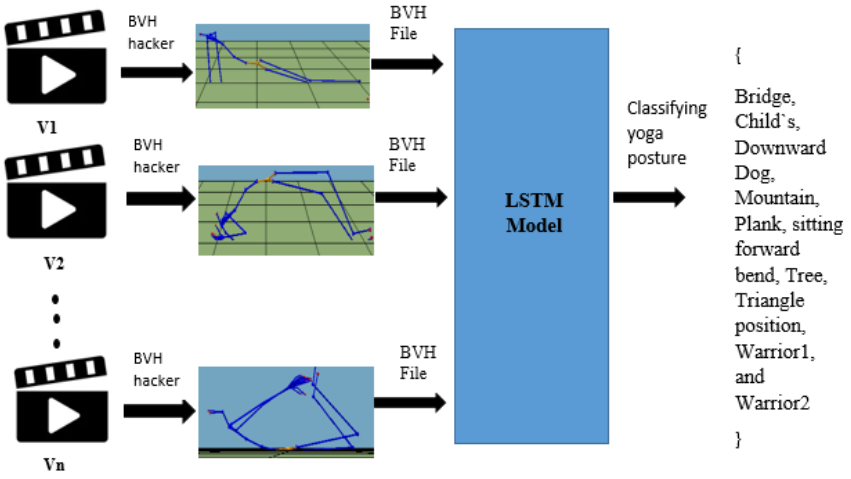


Figure.15. Workflow of the LSTM model

Our training data set comprises of over 10000 frame samples from 250 video frames divided into 10 yoga poses. As demonstrated by our experiments, deeper networks are better able to predict longer animations. The third layer (output) is responsible for encoding the expected

outcome to one of the yoga posture courses outlined in earlier sections. The second layer (stacked LSTM layer) is a comparable LSTM layer. The input layer must equal the size of the output, or the number of potential labels that may be outputted, even though it need not match the size of the input data. Figure 15 shows how the LSTM model works.

5.4.2 BVH motion data structure

The BVH Motion-Capture file is widely used to preserve the data acquired by the Motion-Capture device. We have used BVH file format which is widely used animation applications such as Poser, 3DSMax, BVH hacker, etc. In the first section of the BVH file, the skeletal hierarchy and starting position are defined, and the channel data for each frame is discussed in the second part. According to the statistics, the Root joint has six degrees of freedom as opposed to the three Euler-angles rotations for each non-root joint, which consists of three translations and three Euler-angles rotations (see Figure 16).

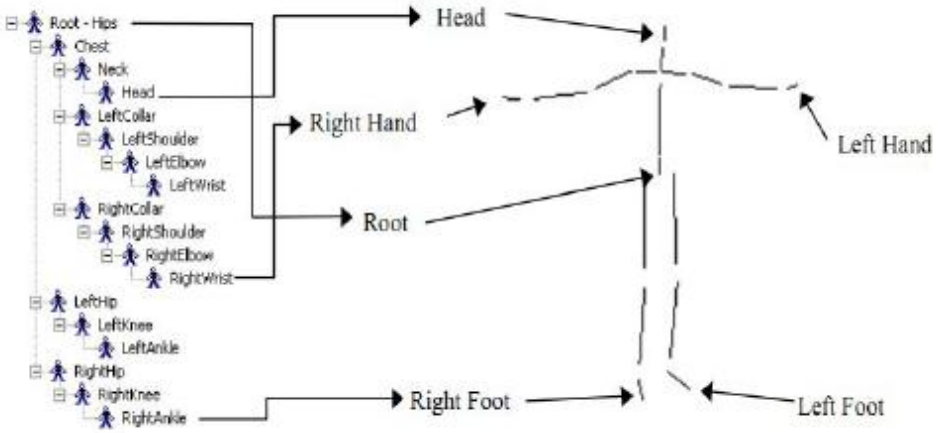


Figure.16.Hierarchical structure for human figure

5.4.3 Experimental analysis

As a technique to better analyse the model's accuracy in our experiments. The content is edited of a series of Motion-Capture recordings to show only one asana per file. All 55 files in this test were divided into ten yoga postures: ‘Bridge’, ‘Child`s’, ‘Downward Dog’, ‘Mountain’, ‘Plank’, ‘Sitting Forward Bend’, ‘Tree’, ‘Triangle Position’, ‘Warrior1’, and ‘Warrior2’. This approach allows for the expression and storage of complex motion data (Figure 17).

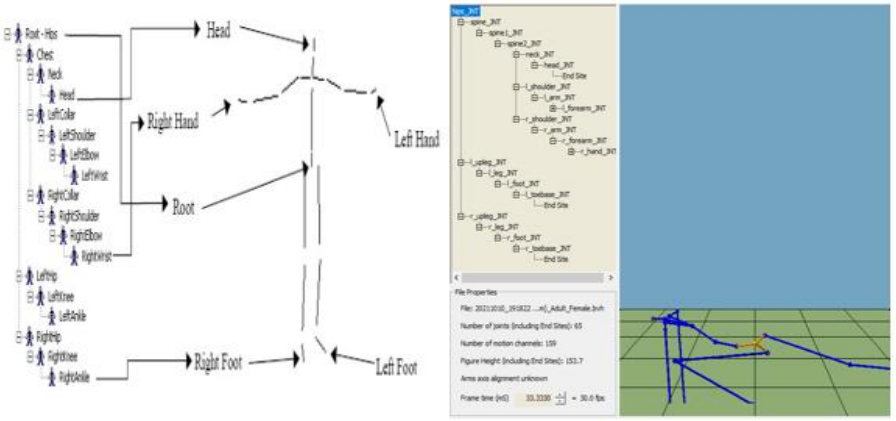


Figure.17. BVH motion data structure

As well as the results obtained by merging all trials using this yoga posture shown in Figure 18.Finally, the results of these investigations are positive, demonstrating the model's longevity and demonstrating that a stacked LSTM Neural-Network (NN) can successfully train to detect yoga pose using Motion-Capture file. Figure 19 below depicts the results for the best experiment performed under the described conditions (sample size = 2404) and also the resulting obtained by combining all ten-yoga posture.

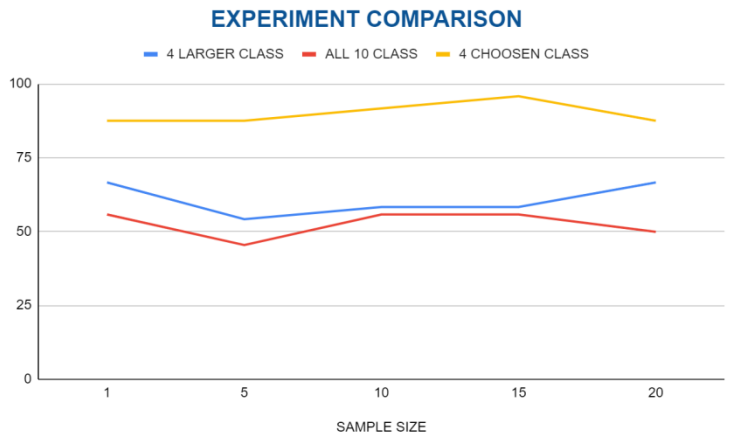


Figure.18. Results of combining 3 experiments used this yoga posture

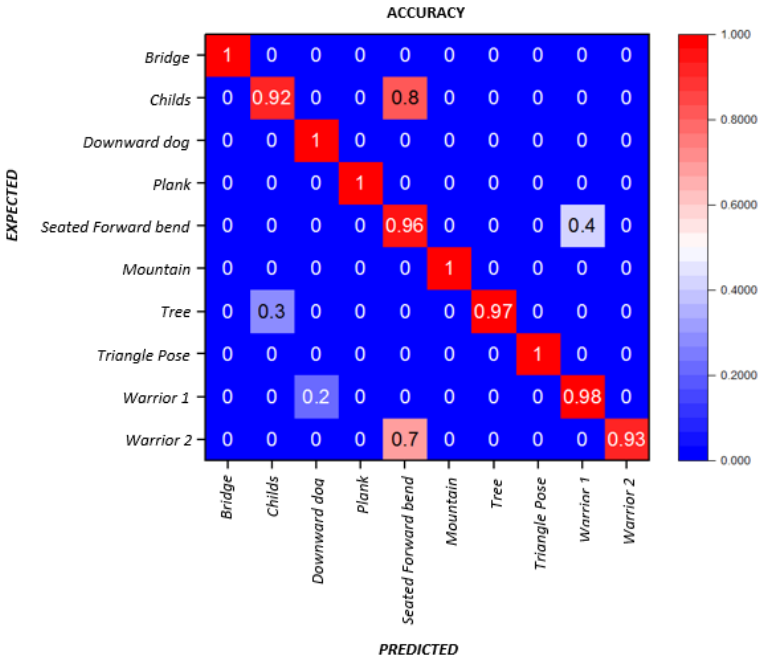


Figure.19.Confusion matrices

6. ORGANIZATION OF THESIS

Chapter-1: Introduction - This first chapter, provides the background of yoga pose estimation for real time. The required technologies for developing such a solution are introduced and the problem statement that this research aims to solve is outlined. The significance of this research and its potential impact on the real time field is also discussed.

Chapter-2: Literature Review - A comprehensive review of the existing literature is presented in this chapter on yoga pose estimation, including various approaches and techniques employed by researchers in their research work and concluded the research gap.

Chapter-3: Predicting Yoga Poses Using Deep Learning Algorithm - Real-time videos are used to extract functions from the key points of every frame, derived from the pose estimation algorithm, and long short-term memory is applied to provide tentative predictions.

Chapter-4: Recognition of spatio-temporal feature learning with deep learning techniques- A novel feature extraction method using Open Pose is described; together with a DenceBi-directional LSTM network to represent spatial-temporal links in both the forward and backward directions. This will increase the efficacy and consistency of modeling long-range action detection.

Chapter-5: Yoga poses recognition using transfer learning with deep representations - This research work presents an innovative method for pose recognition using pre-trained Convolutional Neural Networks (CNNs) model as source architecture for extracting features from the target dataset, followed by a hybrid Support Vector Machines and K-Nearest Neighbor (SVM-KNN) classifier for action classification.

Chapter-6: Detecting Yoga Posture in the Bio vision Hierarchy Using Deep Learning Model- This study describes a real-time method for creating standard motion data files.

Chapter-7: Conclusion and Future work, specifies the research contributions and future research perspectives.

Reference

1. Aggarwal J.; Ryoo M.; Human activity analysis: A survey. *ACM Comput Surv*., (2011),1-43.
2. Poppe R.; A survey on vision-established human action recognition. *Image Vis Comput*, (2010),976-90.
3. S. Yadav, A. Singh, A. Gupta, and J. Raheja, “Real-time yoga recognition using deep learning”, Neural Comput. and Appl.,(2019).
4. U. Rafi, B. Leibe, J.Gall, and I. Kostrikov, “An efficient convolutional network for human pose estimation”, British Mach. Vision Conf., (2016).
5. S. Haque, A. Rabby, M. Laboni, N. Neehal, and S. Hossain, “ExNET: deep neural network for exercise pose detection”, Recent Trends in Image Process. and Pattern Recog., (2019).
6. S. Patil, A. Pawar, A. Peshave, A. N. Ansari, and A. Navada, “Yoga tutor visualization and analysis using SURF algorithm,” in Proceedings of the 2011 IEEE Control and System Graduate Research Colloquium, IEEE, Shah Alam, Malaysia, June (2011),43–46.
7. W. Wu, W. Yin, and F. Guo, “Learning and self-instruction expert system for Yoga,” in Proceedings of the 2010 2nd International Workshop on Intelligent Systems and Applications, IEEE, Mumbai, India,(2010),1–4.
8. H. T. Chen, Y. Z. He, C. L. Chou, S. Y. Lee, B. S. P. Lin, and J. Y. Yu, “Computer-assisted self-training system for sports exercise using kinects,” in Proceedings of the 2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), IEEE, London, UK, (2013),1–4.
9. E. W. Trejo and P. Yuan, “Recognition of Yoga poses through an interactive system with Kinect device,” in Proceedings of the 2018 2nd International Conference on Robotics and Automation Sciences (ICRAS), IEEE, Wuhan, China, (2018), 1–5.
10. A. Mohanty, A. Ahmed, T. Goswami, A. Das, P. Vaishnavi, and R. R. Sahay, “Robust pose recognition using deep learning,” in Proceedings of the International Conference on Computer Vision and Image Processing, Springer, Singapore,(2017),93–105.
11. A. Toshev and C. Szegedy, “Deeppose: human pose estimation via deep neural networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, Columbus, OH, USA,(2014) ,1653–1660.
12. M. Islam, H. Mahmud, F. Ashraf, I. Hossain and M. Hasan, "Yoga posture recognition by detecting human joint points in real time using microsoft kinect", IEEE Region 10 Humanit. Tech. Conf., (2017) ,668-67.

13. S. Patil, A. Pawar, and A. Peshave, “Yoga tutor: visualization and analysis using SURF algorithm”,Proc. IEEE Control Syst. Graduate Research Colloq.,(2011),43-46.
14. P. Dar, “AI guardman – a machine learning application that uses pose estimation to detect shoplifters”. [Online].
15. D. Mehta, O. Sotnychenko, F. Mueller and W. Xu, “XNect: real-time multi-person 3D human pose estimation with a single RGB camera”, ECCV, (2019).
16. A. Gupta, T. Chen, F. Chen, and D. Kimber, “Systems and methods for human body pose estimation”, U.S. patent, (2011), 7,925.
17. H. Sidenbladh, M. Black, and D. Fleet, “Stochastic tracking of 3D human figures using 2D image motion”, Proc 6th European Conf. Computer Vision, (2000).
18. A. Agarwal and B. Triggs, “3D human pose from silhouettes by relevance vector regression”, Intl Conf. on Computer Vision & Pattern Recogn,(2004), 882–888.
19. W. Gong, X. Zhang, J. Gonzàlez, A. Sobral, T. Bouwmans, C. Tu, and H. Zahzah, “Human pose estimation from monocular images: a comprehensive survey”, Sensors, Basel, Switzerland, (2016).
20. A. Mohanty, A. Ahmed, T. Goswami, “Robust pose recognition using deep learning”, Adv. in Intelligent Syst. and Comput, Singapore, (2017),93-105.
21. M. Dantone, J. Gall, C. Leistner, “Human pose estimation using body parts dependent joint regressors”, Proc. IEEE Conf. Computer Vision Pattern Recogn., (2013).
22. G. Ning, P. Liu, X. Fan and C. Zhan, “A top-down approach to articulated human pose estimation and tracking”, ECCV Workshops, (2018).
23. M. Li, Z. Zhou, J. Li and X. Liu, “Bottom-up pose estimation of multiple person with bounding box constraint”, 24th Intl. Conf. Pattern Recogn.,(2018).

LIST OF PUBLICATIONS

Journal Publications

- Palanimeera, J.,Ponmozhi, K.(2023),"Yoga Posture Recognition by Learning Spatial-Temporal Feature with Deep Learning Techniques", Int. J. of Image and Graphics, 2023, 2450055.IF: 1.6.

- Palanimeera, J., Ponmozhi, K.(2023), “YAP_LSTM: yoga asana prediction using pose estimation and long short-term memory” , Soft Computing.IF: 4.1.

- Palanimeera, J., Ponmozhi, K.(2021),"Yoga Posture Recognition According to Image Captured by RGB Camera Turkish journal of physiotherapy and rehabilitation.

- J.Palanimeera , K.Ponmozhi, "Detecting Yoga Posture in the Biovision Hierarchy Using a Deep Learning Model",journal International Journal of Pattern Recognition and Artificial Intelligence, (Under Review).

Conference Publications

- Palanimeera, J., Ponmozhi, K.,(2020)"Classification of yoga pose using machine learning techniques Materials Today: Proceedings", 37(Part 2), pp. 2930–2933.

- Palanimeera, J. , Ponmozhi, K.(2022),"Transfer Learning with Deep Representations is Used to Recognition Yoga Postures" 1st International Conference on Electrical, Electronics, Information and Communication Technologies, ICEEICT 2022.

- Palanimeera, J. , Ponmozhi, K. ,(2020), “ Techniques Used to Capture Skeletal Information and Their Performance Accuracy: A Literature Review.” Proceedings of the 2nd International Conference on IoT, Social, Mobile, Analytics & Cloud in Computational Vision & Bio-Engineering (ISMAC-CVB 2020).

- Palanimeera, J., Ponmozhi, K.(2023),"Pose estimation using machine learning and feature extraction",Mathematics and Computer Science. 1, pp. 405–424. (Book Chapter)